"On the engineers' new toolbox" or Analog Circuit Design, using Symbolic Analysis, Computer Algebra, and Elementary Network Transformations

(2011 Re-Release)

Eberhard H.-A. Gerbracht

***Abstract*—In this paper, by way of three examples – a fourth order low pass active RC filter, a rudimentary BJT amplifier, and an LC ladder – we show, how the algebraic capabilities of modern computer algebra systems can, or in the last example, might be brought to use in the task of designing analog circuits.**

ACM Classification: I.1 Symbolic and algebraic manipulation; J.2 Physical sciences and engineering; G.2.2 Graph theory

Mathematics Subject Classification (2000): Primary 94C05; Secondary 94C15, 68W30, 13P10, 05C85

Keywords: analog circuits, filter design, polynomial equations, computer algebra, delta-wye transformation.

I. Introduction

For those about to embark on the study of this paper, let us start with a warning: This article is not a report on ground-breaking results. Neither will we present spanking new circuits, ripe for patent. Rather in spirit of SM^{2}ACD's explicit policy to be "a forum for the discussion of new ideas and methodologies", we would like to advocate the (re)introduction[1] of computer algebra into the design process of analog circuits. In order to do this, we present some examples, where computer algebra (together with numerics) was used in the design, especially the sizing of the given circuits.

These examples do not represent a systematic approach, but they should be considered as tentative steps towards getting to know the capabilities (and some of the shortcomings) of computer algebra systems (CAS), and making these tools bring forth fruit in the daily work of electrical engineers.

II. The general ansatz

One of the principal problems in analog circuit design[2] is to determine element values in a given circuit in such a way that the resulting network function is in accordance with some previously given performance specifications. In more mathematical terms: suppose the network function is demanded to be a rational function of form

$$f_{des}(s) = \frac{a_0 + a_1 s + \cdots + a_m s^m}{b_0 + b_1 s + \cdots + b_n s^n}, \tag{1}$$

with the numbers $m, n, a_1, \ldots, a_m, b_1, \ldots, b_n$ determined by the initial specifications.

Then we need to find an analog circuit Γ built from elements, which themselves are described by parameters $p_1, \ldots, p_k$, such that its network function

$$H_\Gamma(s) = \frac{\sum_{\mu=0}^{m} A_\mu(p_1, \ldots, p_k) s^i}{\sum_{\nu=0}^{n} B_\nu(p_1, \ldots, p_k) s^i}, \tag{2}$$

where the terms A_μ and B_ν are polynomials in $p_1, \ldots, p_k$, satisfies

$$H_\Gamma(s) = f_{des}(s). \tag{3}$$

In other words: at the end of the day, many design problems result in a system of nonlinear equations

$$\begin{aligned} A_\mu(p_1, \ldots, p_k) - a_\mu &= 0, \\ B_\nu(p_1, \ldots, p_k) - b_\nu &= 0, \end{aligned} \tag{4}$$

$1 \le \mu \le m$, and $1 \le \nu \le n$ for variables $p_1, \ldots, p_k$.
If indeed the terms A_μ, B_ν are polynomials[3] with coefficients in some number field, the set of solutions, otherwise also known as the zero set of (4), forms what is called an *affine algebraic variety*, an object which belongs to the mathematical subdiscipline of *Algebraic Geometry*. Thus when trying to solve these equations, we could profit both from the centuries of experience, mathematicians have gained in this particular field, and from the more recent results and algorithmic instruments developed – the concept of *Gröbner bases*, as well as the *Buchberger algorithm* and its "relatives" which are realized in nearly all of the current computer algebra systems. The result would be exact ("symbolic") descriptions of the sought after solution sets, from which we could deduce numeric results afterwards.
Now, when we take a look at the advances in symbolic methods applied to circuit design, we can make a peculiar observation: nowadays it has become nearly routine to calculate any given network function as a multivariable rational function of the complex frequency s and the symbolic circuit elements (even though there is still the problem of the combinatorial

This article first appeared in: SM^{2}ACD'08 Proceedings of the Xth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design, Erfurt, Germany, October 07 to 08, 2008; pp. 127-134. Due to the low distribution of these proceedings, the author has decided to make the article available to a larger audience through the arXiv.

The author's current (as of December 31st, 2010) address is Bismarckstraße 20, D-38518 Gifhorn, Germany. Current e-mail: `e.gerbracht@web.de`

[1] **Re**introduction, because some symbolic analysis tools were and still are closely associated to existing computer algebra systems, although nowadays they do not make much use of the advanced algebraic capabilities (it is debatable if they ever did), but rely more on their numeric proficiencies.

[2] See e.g. [1].

[3] Thus for example we do not talk about the problem of biasing a transistor, since then we would have to solve equations containing the exponential function (as in the Ebers-Moll model of the BJT), or equations which contain roots, and moreover are piecewisely defined (as for FETs).

explosion for large circuits). However, when it comes to finding solutions for the nonlinear equations arising from the task of sizing a particular circuit, even if it is relatively small, researchers, as well as designers, fall back upon numerical means and various approximative methods with their various drawbacks.

III. An Example - Sizing an active 4th order low pass RC filter

In section 6.3.3. of [2] the following filter circuit was presented:

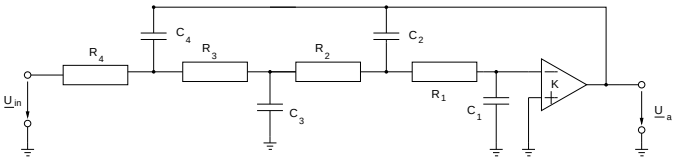

Fig. 1. Circuit topology for fourth order low pass RC filter

The aim there was to construct a fourth order Butterworth filter with this topology.

Using one of the available symbolic analysis programs[4], we see that the circuits symbolic transfer function is given by

$$H(s) := \frac{\underline{U}_a}{\underline{U}_{in}} = \frac{K}{D(s)} \tag{5}$$

with $D(s) = 1+a_1 s+a_2 s^2+a_3 s^3+a_4 s^4$ being a polynomial, and

$$\begin{aligned}
a_1 &= ((1-K)(C_4+C_2)+C_3+C_1)R_4+((1-K)C_2+C_3+C_1)R_3+\\
&\quad ((1-K)C_2+C_1)R_2+C_1R_1,\\
a_2 &= C_2C_1R_1R_2+(C_3+C_2)C_1R_1R_3+(C_4+C_3+C_2)C_1R_1R_4+\\
&\quad C_3((1-K)C_2+C_1)R_2R_3+((C_4+C_3)((1-K)C_2+C_1)R_2R_4+\\
&\quad C_4(C_3+(1-K)C_2+C_1)R_3R_4,\\
a_3 &= C_3C_2C_1R_1R_2(R_3+R_4)+C_4C_2C_1R_1(R_2+R_3)R_4+\\
&\quad C_4C_3C_1(R_1+R_2)R_3R_4+(1-K)C_4C_3C_2R_2R_3R_4,\\
a_4 &= C_4C_3C_2C_1R_1R_2R_3R_4.
\end{aligned} \tag{6}$$

In order to specify element values in such a way that $H(s)$ becomes the transfer function of a fourth order Butterworth filter, we can proceed from these data by matching the coefficients with those numerical values, which can be found in any table for filter design, and then try to solve the resulting equations by numerical means. Since we only have four equations for nine unknowns, there is the need to reduce the existing degrees of freedom by more or less arbitrarily attaching numerical values to some of the element values. In fact, that is what was done in [2], where K was set equal to 2 and all the C_i were set to 1 (all element values in this example are considered to be normalized); afterwards some variant of the Newton-Raphson method was used, to deduce one single solution.

The disadvantages of such an approach are obvious: Using rounded values (from precalculated tables) introduces errors right from the start. Both, the assignment of (quite arbitrary) numbers to arbitrarily chosen parameters, and the choice of a starting vector for the Newton-Raphson method might hinder from finding a solution, or even worse, might make a solution impossible[5]

Furthermore, questions like "Do alternatives exist? How many are there?", cannot be answered either, and changing any of the numerical parameters forces us to redo most of the calculations.

When we tried to solve this design task, first of all we started with a slightly different approach, using the Feldtkeller equation

$$D(s) \cdot D(-s) = 1 + s^8 \tag{7}$$

for the special case of the fourth order Butterworth filter. Thus we got rid of the precalculated approximate values for the a_i, by using instead the defining algebraic equations given by matching the coefficients from

$$\begin{aligned}
D(s) \cdot D(-s) &= 1 + (2a_2 - a_1^2)s^2 + (a_2^2 - 2a_1a_3 + 2a_4)s^4\\
&\quad + (2a_2a_4 - a_3^2)s^6 + a_4^2 s^8
\end{aligned}$$

with $1 + s^8$. Consequently we enlarged the set of equations (6) by adding

$$2a_2 - a_1^2 = 0,\ a_2^2 - 2a_1a_3 + 2a_4 = 0,\ 2a_2a_4 - a_3^2 = 0,\ a_4 = 1 \tag{8}$$

to it, and regarding $a_1, \ldots, a_4$ as further unknowns.

Even though in this paper we propose the usage of the algebraic capabilities of a CAS, at this point we used Mathematica (Version 6.0.1.0) as a numerical engine – not in order to approximate one solution, but to find approximates for **all** solutions. Moreover as was done in [2], we also fix $K = 2$, and $C_i = 1$ for $1 \leq i \leq 4$. Since obviously the enlargement of the set of equations results in a bigger set of solutions, we were left with the final task of "post-processing" this zero set, by singling out those particular solutions for which $a_1, \ldots, a_4$ determine a polynomial $D(s)$, all of whose zeroes have negative real part (= filter stability), and such that all the R_i are *non-negative* (= realizability by classic elements). Indeed, we observe that from all the possible solutions calculated, only nine satisfy the restrictions concerning the a_i, and again only one of these,

$$\begin{aligned}
K &= 2,\ C_1 = C_2 = C_3 = C_4 = 1,\\
R_1 &\approx 0.13393381829719465263108758009 0,\\
R_2 &\approx 3.89303669731839240287174614900 6,\\
R_3 &\approx 2.47919211145555840308219876669 6,\\
R_4 &\approx 0.77359039853632997704317592784 1.
\end{aligned}$$

results in the R_i all being positive.

So, where does computer algebra enter this picture?

First of all Mathematica's `NSolve`-procedure, which we used in the above task, at its core heavily depends on algebraic algorithms: the fact that it is able to calculate *all* solutions stems from the algebraic variety, which was defined by (6) and (8), being *zero dimensional*. That is, it consists of finitely many

[4]For the analysis of this and all other circuits within this paper, we use SapWin 3.0, because of its affordability and its easy availability through the world wide web [3].

[5]E.g. this is the case, when we use the preassigned values from [2], and try to find element values, such that the above circuit attains the characteristics of a fourth order Chebyshev filter.

points. Again we can be sure of this fact, only because there is an algorithm – the *Buchberger algorithm*[6] – which is used by Mathematica as a subprocedure of `NSolve` and which for any system of algebraic equations produces particularly well-behaved equivalent systems of equations – so-called *Gröbner bases* – which allow much more insight into the properties of the zero set, and may be used for a subsequent elimination of variables[7].

Thus e.g., when we apply Mathematica's `GroebnerBasis`-command to (6) and (8), with the values of $K, C_1, \ldots, C_4$ fixed as above, and the variables ordered lexicographically by $a_1 < \ldots a_4 < R_1 < \cdots < R_4$ (which amounts to saying that first a_1 is to be eliminated, then a_2, etc.), then the resulting Gröbner basis contains the polynomial

$$\begin{aligned}
&P = 1 - 160\,R_4^4 + 13872\,R_4^8 - 788512\,R_4^{12} + 31505120\,R_4^{16}\\
&-920274816\,R_4^{20} + 20065991808\,R_4^{24}\\
&-328437088768\,R_4^{28} + 4000414289152\,R_4^{32}\\
&-35535204282368\,R_4^{36} + 223781766674432\,R_4^{40}\\
&-956822102532096\,R_4^{44} + 2535921430958080\,R_4^{48}\\
&-3050522934050816\,R_4^{52} - 2341746368053248\,R_4^{56}\\
&+10182414501412864\,R_4^{60} - 1806331484831744\,R_4^{64}\\
&-13996296348106752\,R_4^{68} + 2888816545234944\,R_4^{72}
\end{aligned}$$

in R_4 of degree 72, which implies that there are only finitely many possible values for R_4. Furthermore this Gröbner basis contains polynomials $p_1(R_1, R_4)$, $p_2(R_2, R_4)$, and $p_3(R_3, R_4)$, each of which being linear in the respective R_i, and otherwise consisting only of a high degree polynomial in R_4. Since these polynomials as part of a Gröbner basis have to be zero, too, consequently each zero of P determines precisely one solution vector $(R_1, \ldots, R_4)$.

Those solutions, which result in the additional specifications for the a_i to be fulfilled, originate from a factor of the polynomial P of degree 18, which can be calculated using Mathematica's `Factor`-procedure, when we allow the coefficients to be sums of integers and integer multiples of $\sqrt{2}$. This factor polynomial is

$$\begin{aligned}
&-10 + 7\sqrt{2} + (-12 + 8\sqrt{2})R_4^2 + (280 - 196\sqrt{2})R_4^4\\
&+(520 - 352\sqrt{2})R_4^6 + (-2824 + 1900\sqrt{2})R_4^8\\
&+(-10816 + 6656\sqrt{2})R_4^{10} + (-13904 + 7368\sqrt{2})R_4^{12}\\
&+(-8288 + 5952\sqrt{2})R_4^{14} + (1280 + 6592\sqrt{2})R_4^{16}\\
&+10368R_4^{18}.
\end{aligned}$$

Substituting $T = R_4^2$, we see that its structure is effectively determined by a polynomial of degree 9, which results in the number of admissable solutions being nine.

[6]The Buchberger algorithm can be seen as a generalization of both the Euclidean algorithm (to the multidimensional case) and the Gauß algorithm (to polynomial equations of degree > 1).

[7]In most cases, especially if the zero set is zero dimensional, there exist Gröbner bases which are in *triangular form*, a property which generalizes the analogous property of sets of linear equations, which can be brought to triangular form using the Gauß algorithm. For more details see [4], [5].

To finish this section, let us remark that by modifying the Feldtkeller equation to

$$D(s) \cdot D(-s) = 1 + T_4(s) \cdot T_4(-s),$$

where $T_4(s)$ denotes the fourth order Chebyshev polynomial, and slightly changing the initial values for the C_i, after some trial-and-error we were able to calculate the following set of parameter values, with which we can make the initially given circuit into a Chebyshev filter

$$\begin{aligned}
K &= 2,\ C_1 = 1,\ C_2 = 2,\ C_3 = 2,\ C_4 = 1,\\
R_1 &\approx 0.26363809085479418546159378 7592,\\
R_2 &\approx 0.62416476544787900031652578 6179,\\
R_3 &\approx 2.64518522675854531274475059 2839,\\
R_4 &\approx 3.24901339987364963343777745 1232.
\end{aligned}$$

The question remains for each filter characteristic: which set of prescribed values for $K, C_1, \ldots, C_4$ imply solutions for $R_1, \ldots, R_4$, which make sense in an electrical engineering point of view, i.e., for which positive $K, C_1, \ldots, C_4$ do we get positive $R_1, \ldots, R_4$?

IV. Another Example: Sizing a BJT Amplifier with respect to break frequencies

From [6] - a students' textbook in analog electronics - we take our next example: a rudimentary BJT amplifier circuit with only a few additional classic elements for biasing and sizing purposes (see figure (IV)). The biasing resistances are

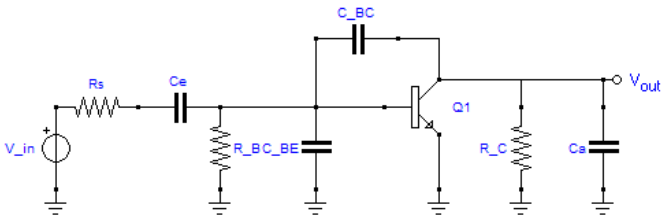

Fig. 2. Rudimentary topology for a BJT amplifier

given by $R_B = 310\text{k}\Omega$, and $R_C = 2\text{k}\Omega$. The biased transistor is modelled by concrete h-parameters

$$H = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}, = \begin{pmatrix} 672\Omega & 0 \\ 96 & 35 \cdot 10\mu\text{S} \end{pmatrix} \qquad (9)$$

and its capacitances $C_{BC} = 11\text{pF}$, and $C_{BE} = 320\text{pF}$.

With denoting its transfer function by $H(s) = \frac{N(s)}{D(s)}$, a full symbolic analysis with SapWin 3.0. gives

$$N(s) = C_e h_{21} R_B R_C \cdot s - C_{BC} C_e h_{11} R_B R_C \cdot s^2,$$

and $D(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3$, with

$$\begin{aligned}
a_0 &= -h_{11} - R_B + h_{12}h_{21}R_C - h_{11}h_{22}R_C - h_{22}R_B R_C,\\
a_1 &= -C_{BC}h_{11}R_B - C_{BE}h_{11}R_B - C_e h_{11}R_B - C_a h_{11}R_C\\
&\quad -C_{BC}h_{11}R_C - C_a R_B R_C - C_{BC}R_B R_C + C_{BC}h_{12}R_B R_C\\
&\quad -C_{BC}h_{21}R_B R_C + C_{BC}h_{12}h_{21}R_B R_C + C_{BE}h_{12}h_{21}R_B R_C\\
&\quad +C_e h_{12}h_{21}R_B R_C - C_{BC}h_{11}h_{22}R_B R_C - C_{BE}h_{11}h_{22}R_B R_C\\
&\quad -C_e h_{11}h_{22}R_B R_C - C_e h_{11}R_s - C_e R_B R_s + C_e h_{12}h_{21}R_C R_s\\
&\quad -C_e h_{11}h_{22}R_C R_s - C_e h_{22}R_B R_C R_s,\\
a_2 &= -C_a C_{BC}h_{11}R_B R_C - C_a C_{BE}h_{11}R_B R_C - C_{BC}C_{BE}h_{11}R_B R_C\\
&\quad -C_a C_e h_{11}R_B R_C - C_{BC}C_e h_{11}R_B R_C - C_{BC}C_e h_{11}R_B R_s\\
&\quad -C_{BE}C_e h_{11}R_B R_s - C_a C_e h_{11}R_C R_s - C_{BC}C_e h_{11}R_C R_s\\
&\quad -C_a C_e R_B R_C R_s - C_{BC}C_e R_B R_C R_s + C_{BC}C_e h_{12}R_B R_C R_s\\
&\quad -C_{BC}C_e h_{21}R_B R_C R_s + C_{BC}C_e h_{12}h_{21}R_B R_C R_s\\
&\quad +C_{BE}C_e h_{12}h_{21}R_B R_C R_s - C_{BC}C_e h_{11}h_{22}R_B R_C R_s\\
&\quad -C_{BE}C_e h_{11}h_{22}R_B R_C R_s,\\
a_3 &= -C_a C_{BC}C_e h_{11}R_B R_C R_s - C_a C_{BE}C_e h_{11}R_B R_C R_s\\
&\quad -C_{BC}C_{BE}C_e h_{11}R_B R_C R_s.
\end{aligned}$$

With regard to the concretely given transistor parameters at the bias point, the nominator and denominator of $H(s)$ become

$$N'(s) = (290625000000 C_e s(-1000000000000 + 77s)),$$

and

$$\begin{aligned}
D'(s) &= 1623139843750000000\\
&\quad +3590505265625\, s\\
&\quad +30339062500000000000000\, C_a s\\
&\quad +10883906250000000000000\, C_e s\\
&\quad +1623139843750000000\, C_e R_s s\\
&\quad +7161\, s^2\\
&\quad +673378125000000\, C_a s^2 + 22378125000000\, C_e s^2\\
&\quad +203437500000000000000000 C_a C_e s^2\\
&\quad +3590505265625 C_e R_s s^2\\
&\quad +303390625000000000000000 C_a C_e R_s s^2\\
&\quad +7161 C_e R_s s^3 + 673378125000000 C_a C_e R_s s^3,
\end{aligned}$$

respectively. Thus, e.g. setting $R_s = 0\Omega$, $C_a = 300$pF, and $C_b = 0.1\mu$F, we get

$$H(s) = \frac{465000s(-1000000000000+77s)}{25970237500000000000+1813435834250000s+1015651791s^2}, \tag{10}$$

the poles being (approximately[8])

$$p_1 \approx -14437.758, \quad \text{and} \quad p_2 \approx -1,771051.964.$$

This results in the Bode plot shown in figure 3.

We will now use this example as a problem in circuit design. Our task shall be: with the bias point remaining unchanged, find parameters R_s, C_a, and C_e in such a way that a new transfer function $\widetilde{H}(s)$ is achieved, which has the same zeroes

[8]Clearly, due to the nominator being a quadratic function, we would also be able to calculate exact values.

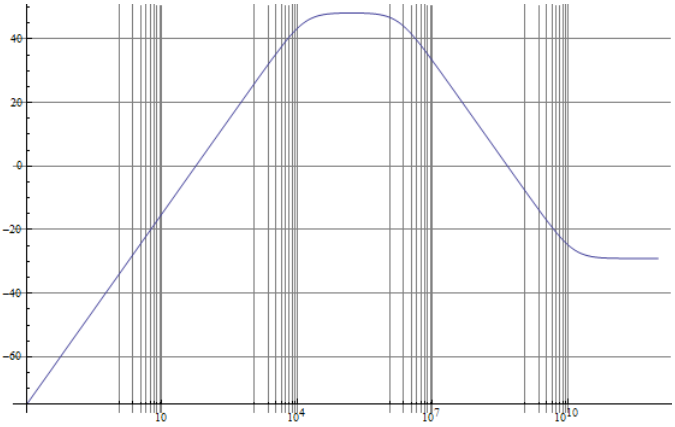

Fig. 3. Bode Plot of initially given BJT amplifier, according to (10)

as $H(s)$, and possesses two (new) poles p_1 and p_2. Introducing a new parameter k (since due to the simplicity of our design topology, we cannot specify the maximal attenuation of $\widetilde{H}$, as we will see below), we thus are asked to determine the parameters k, R_s, C_a, and C_e in such a way that

$$\frac{N'(s)}{D'(s)} = \widetilde{H}(s) = \frac{k \cdot s \cdot (77s - 1000000000000)}{(s-p_1)\cdot(s-p_2)}. \tag{11}$$

This is equivalent to

$$N'(s)\cdot(s-p_1)\cdot(s-p_2) - D'(s)\cdot k\cdot s\cdot(77s - 1000000000000) = 0. \tag{12}$$

Factoring out common terms on both sides of the $-$-sign, we are left with a third order polynomial in s with coefficients being integer polynomials in the variables $\{k, R_s, C_a, C_e, p_1, p_2\}$. To solve (12) we need to set these coefficients to zero, which consequently results in the following system of four algebraic equations[9] (the i-th equation resulting from the coefficient of s^i):

$$\begin{aligned}
0 &= 1623139843750000000\, k - 290625000000\, C_e p_1 p_2,\\
0 &= 3590505265625\, k + 303390625000000000000000\, C_a k\\
&\quad + 10883906250000000000000\, C_e k + 290625000000\, C_e p_1\\
&\quad + 290625000000\, C_e p_2 + 1623139843750000000\, C_e k R_s,\\
0 &= -290625000000\, C_e + 7161\, k\\
&\quad + 673378125000000\, C_a k + 22378125000000\, C_e k\\
&\quad + 203437500000000000000000\, C_a C_e k\\
&\quad + 3590505265625\, C_e k R_s\\
&\quad + 303390625000000000000000\, C_a C_e k R_s,\\
0 &= 7161\, C_e k R_s + 673378125000000\, C_a C_e k R_s.
\end{aligned} \tag{13}$$

Alhough this system is nonlinear, using its procedure `Solve`, Mathematica is able to calculate four symbolic solutions for k, R_s, C_a, and C_e as functions of p_1 and p_2, voicing the explicit caveat: "`Equations may not give solutions for all "solve" variables.`". Nevertheless, indeed there are only four different solutions.

[9]This is why we had to introduce the additional parameter k, since we consider p_1 and p_2 to be fixed.

The first is irrelevant for the concrete problem at hand from an electrical engineering point of view, since it is given by

$$C_e = 0 \quad \text{and} \quad k = 0, \tag{14}$$

and thus results in a trivial transfer function.

The second one, though it leads to nontrivial terms for C_e, k and R_s, furthermore implies

$$C_a = -\frac{11}{1034375000000}, \tag{15}$$

which again has to be considered absurd from an engineering point of view[10].

Finally, there are two solutions left which lead to two different valid sets of design parameters, both of them having in common that $R_s = 0$.

Here we will not present the explicit formulas for C_a, C_e, k, but we will give a numerical example, and we will further show, how the formulas for C_a (and beyond this our claim, that there are only four solutions) can be derived with the help of Gröbner bases.

First, for the numerical example let $p_1 = -10$, and $p_2 = -1000$. Then the two remaining approximate solutions of the above set of equations are

$$(C_a, C_e, k) \approx (53.4989\mu\text{F}, 1.49102\mu\text{F}, 2.66969 \cdot 10^{-9}) \tag{16}$$

and

$$(C_a, C_e, k) \approx (53.5000\mu\text{F}, 149.12881\mu\text{F}, 2.67017 \cdot 10^{-7}). \tag{17}$$

Indeed, the corresponding Bode plots for the respective circuits can be seen in figure 4.

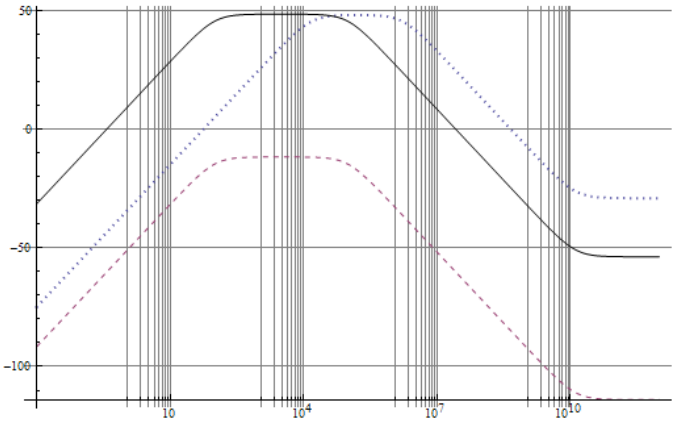

Fig. 4. Bode plot of BJT amplifier circuit with original parameters (dotted), parameters (16) (dashed), and (17) (thick)

Finally, applying Mathematica's `GroebnerBasis`-command to the right-hand sides of equations (13), with the order of variables chosen to be $k < R_s < C_a < C_e$, results in a Gröbner basis consisting of 11 polynomials with exactly one of them containing only the variables C_e, p_1, p_2. This particular polynomial is of degree four in the variable C_e. Keeping in mind that the zero set of the original polynomials constituting (13) is equal to the zero set of its Gröbner basis, we thus see that there truly are only four possibilities for the values of C_e.

Factoring this particular polynomial (again with the help of Mathematica) we get the following factors:

$$C_e,$$

and

$$\begin{aligned} &11823003354909708025000000000000000 \\ &- 25918345728188286341500000000\, p_1 \\ &- 25918345728188286341500000000\, p_2 \\ &+ 5681810493667293811609\, p_1 p_2 \\ &+ 17374037139970114590000000000000\, p_1 p_2\, C_e, \end{aligned} \tag{18}$$

and finally

$$\begin{aligned} &10085282030750000000000000000 \\ &- 22384370986950000000\, p_1 \\ &- 22384370986950000000\, p_2 \\ &+ 49070938130697\, p_1 p_2 \\ &- 67626498450000000000000000000\, p_1 C_e \\ &- 67626498450000000000000000000\, p_2 C_e \\ &+ 16321412003400000000000\, p_1 p_2\, C_e \\ &+ 45346707000000000000000000000000\, p_1 p_2\, C_e^2. \end{aligned} \tag{19}$$

Further analysis shows that $C_e = 0$ results in $k = 0$, and the solution of the linear (with respect to C_e) equation implies the negative value of C_a according to (15). Thus the two solutions, whose existence we claimed above, have their origin in the quadratic equation (19).

V. What If equations become too complicated? An Outlook

Symbolic methods always suffer from one major deficiency – the "combinatorial explosion" of terms – which most of the time results in both, a massive, nearly insatiable demand for space, and an extraordinary high amount of computing time. This observation also applies to all Gröbner basis methods: It is known that by necessity the problem of finding a Gröbner basis in the general case needs an exponential amount of space[11]. On the other hand, in recent years there have been major improvements with regard to the run-time behaviour of the original Buchberger algorithm[12], which have resulted in faster and thus more powerful algorithms to compute Gröbner bases.

As we could see in the above examples, even moderately sized small circuits already give rise to mid-sized equations[13], which again – by way of the methods from computer algebra – result in even bigger symbolic expressions for the sought-after

[10] Unless we allow for NICs (negative impedance converters) to be added to the initial circuit topology.

[11] For more details and/or a mathematically more precise formulation of this statement, see [7].

[12] See e.g. the results by J.-C. Faugére in [8] and [9].

[13] Indeed we must not regard the results, which have been presented up until now, as *large* – the maximal computing time used in the above examples in the worst case, which was the calculation of all the solutions of the Chebyshev RC filter, was ten minutes.

design parameters. Thus the "new toolbox" computer algebra, the usage of which I try to recommend with this article, seems to be doomed from the start.

Still, bad behaviour *in general* does not mean that the instruments from computer algebra *always* behave badly – there might be classes of problems for which results are computed fast, or can be written down very economically. As electrical engineers know, there are close (but in the author's opinion by far not completely understood) connections between a circuit's topology and the algebraic structure of its network function, which may be taken advantage of with regard to design tasks.

Take on the other hand researchers in symbolic methods, who are accustomed with the phenomenon of swelling expressions arising in the analysis of circuits. As a workaround a number of them have proposed so-called "sequences of expressions"[14]. It is interesting to note that even electrical engineers without experience in methods of symbolic analysis may already be well versed in this "sequential approach", if only they knew classic filter theory, in particular the Cauer canonical forms of LC-circuits.

In form of an example, let us show, what we mean. Suppose we are given an LC-ladder network, consisting of three inductances and three capacitances, as shown in figure 5.

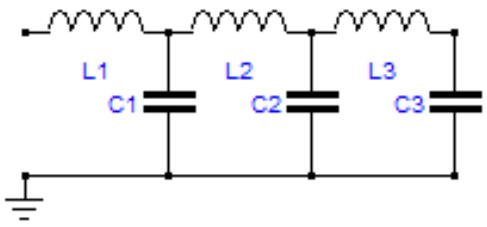

Fig. 5. LC ladder in Cauer canonical form

As is well known its impedance $Z(s)$ can be described by a continued fraction

$$Z(s) = L_1\, s + \cfrac{1}{C_1\, s + \cfrac{1}{L_2\, s + \cfrac{1}{C_2\, s + \cfrac{1}{L_3\, s + \cfrac{1}{C_3\, s}}}}}. \qquad (20)$$

That, in fact, is just short hand for a sequence of expressions. This can be seen by deducing (20) with a process of alternatively combining impedances in series and in parallel. Thus we have

$$Z(s) = L_1\, s + Z_1(s),$$

where $Z_1(s)$ denotes the impedance of the ladder circuit, with L_1 having been removed. Further:

$$\frac{1}{Z_1(s)} = \frac{1}{\frac{1}{C_1\, s}} + \frac{1}{Z_2(s)} = C_1\, s + \frac{1}{Z_2(s)}, \qquad (21)$$

where $Z_2(s)$ is the impedance of the circuit, that is left, when L_1 and C_1 have been removed. Continuing like this, taking away one more circuit element in each step, this process can be repeated, until we get to one last impedance $Z_5(s)$ (which would just consist of C_3).

Reinserting the various Z_i we get

$$\begin{aligned} Z(s) &= L_1\, s + Z_1(s) \\ &= L_1\, s + \frac{1}{C_1\, s + \frac{1}{Z_2(s)}} \\ &= \ldots, \end{aligned}$$

and thus are able to calculate $Z(s)$.

When conversely one wants to realize an LC transimpedance function as an LC ladder, classic theory[15] tells us to expand the given rational function into a continued fraction (an algorithm which in case of rational functions always comes to an end after a finite number of steps) and read off the element values directly from this expression[16], which according to theorems in classic filter theory all are positive.

If we want to bring together the capabilities of a CAS for solving algebraic equations and the sequence of expression approach in our example, we first observe that we can easily calculate the form of the intermediate impedances Z_i, i.e. we have

$$\begin{aligned} &Z_5(s) = \frac{1}{C_3\, s} = \frac{1}{*s}, \quad Z_4(s) = L_3\, s + Z_5(s) = \frac{*s^2 + 1}{*s}, \\ &\frac{1}{Z_3(s)} = C_2\, s + \frac{1}{Z_4(s)} = \frac{*s^3 + *s}{*s^2 + 1}, \ldots, \\ &Z_1(s) = \frac{a_4 s^4 + a_2 s^2 + 1}{a_5 s^5 + a_3 s^3 + a_1 s}, \quad \text{and finally} \\ &Z(s) = L_1\, s + Z_1(s) \\ &\qquad = \frac{(a_5 \cdot L_1) s^6 + (a_3 \cdot L_1 + a_4) s^4 + (a_1 \cdot L_1 + a_2) s^2 + 1}{a_5 s^5 + a_3 s^3 + a_1 s} \end{aligned}$$

with $*$ and the a_i denoting polynomials in the circuit parameters C_1, C_2, C_3, L_2, L_3.

A sensible design task would now be to specify an impedance function

$$Z_{des}(s) = \frac{k\,(A_6 s^6 + A_4 s^4 + A_2 s^2 + 1)}{A_5 s^5 + A_3 s^3 + A_1 s}, \qquad (22)$$

with real numbers $k, A_1, \ldots, A_6$, and to demand $Z(s) = Z_{des}(s)$. Analogously to the above example of the BJT amplifier, by matching coefficients, this results in six nonlinear equations, which have to be solved for the variables $a_1, \ldots, a_5$ and L_1. Here we only present the (easy) solution (determined by Mathematica again) for the first step, which is

$$\begin{aligned} &a_2 = A_2 - \frac{A_1 \cdot A_6}{A_5}, \quad a_4 = A_4 - \frac{A_3 \cdot A_6}{A_5}, \\ &a_5 = \frac{A_5}{k}, \quad a_1 = \frac{A_1}{k}, \quad a_3 = \frac{A_3}{k}, \\ &L_1 = \frac{A_6 \cdot k}{A_5}. \end{aligned}$$

[14] See [10].

[15] See e.g. [11].

[16] Again, nowadays a computer algebra system might be of help for calculating this expansion without much effort, see [12].

To calculate further element values, this procedure needs to be repeated for the other Z_i. Thus one has to solve next the equations resulting from $\frac{1}{Z_1(s)} = C_1\, s + \frac{1}{Z_2(s)}$, where $Z_1(s)$ should still be described in terms of the coefficients a_i, and $Z_2(s)$ is given by the ansatz

$$Z_2(s) = \frac{b_4\, s^4 + b_2\, s^2 + 1}{b_3\, s^3 + b_1\, s} \tag{23}$$

with new unknowns $b_1, \ldots, b_4$. This has to be continued until we have equations for each set of coefficients of Z_i in terms of the coefficients appearing in Z_{i-1}, plus additional equations for the element values. Starting from those equations, successive reinserting leads to the desired expressions for the element parameters.

One has to concede that – compared to the traditional continued fraction expansion – this procedure looks very cumbersome. But in fact, if one looks a little bit closer, it is equivalent to repeatedly doing polynomial division as part of a continued fraction expansion. Furthermore our approach might be generalized to other impedance functions which need not come from ladders, but still ultimately result from a sequence of series, respectively parallel reductions applied to the initially given circuit, i.e. the process of replacing two impedances in series, or in parallel by a single equivalent impedance[17]. Beyond this there is a further generalization to handle Delta-Wye and Wye-Delta transformations.

On the other hand by results from Epifanov and Truemper, it is known that every planar two terminal graph can be reduced to a single edge by these four kinds of transformations (plus removing loops and pendant edges). Moreover there is an algorithm, developed by Feo and Provan[18], which calculates the sequence of necessary graph transformations. Since to each of these transformations there corresponds an equation, which gives the value of the resulting element(s) in terms of the values of previously existing elements, there always is a *sequence of equations* which leads from the initially given element values to the driving point impedance of a planar two terminal circuit[19].

Circuit design for (midsized to large) planar two terminal circuits, which consist of impedances only, thus becomes the task of "reversing" these equations, in an analogous manner to what we did with LC ladders.

The author of this paper firmly believes, that computer algebra systems will be a valuable tool in tackling this particular problem. The challenge remains to find classes of circuits for which the *sequence of reversed equations* can be effectively calculated, or to prove – what might be also possible, and would be an interesting result in itself – that, besides the classic, continued- or partial-fraction-topologies attached to the names of Cauer and Foster, no such further classes exist.

[17]However we need to be careful with the claim for generality, since if we combine e.g. two resistances R_1, R_2 in series to a single resistance R, from the resulting equation $R = R_1 + R_2$, when only given the value of R, we cannot deduce uniquely the values of the R_i.

[18]As a reference for these results in graph theory, see the original paper [13].

[19]This algorithm has been successfully implemented in a student's minor thesis [14] at the TU Braunschweig in 2003.

REFERENCES

[1] Georges Gielen and Willy Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer Academic, Boston, 1991.

[2] Tran chi Hieu, *Anwendung symbolischer Methoden beim Entwurf linearer Analogschaltungen*, Ph.D. thesis, Technische Universität Braunschweig, 2002.

[3] University of Florence. Department of Electronics and Telecommunications, "Sapwin 3.0 - Symbolic Analysis Program for Window," `http://cirlab.det.unifi.it/Sapwin/`, viewed September 1, 2008.

[4] David Cox, John Little, and Donal O'Shea, *Ideals, Varieties, and Algorithms*, Springer, New York, [2]1997.

[5] David Cox, John Little, and Donal O'Shea, *Using Algebraic Geometry*, Springer, New York, [2]2004.

[6] Uwe Naundorf, *Analoge Elektronik*, Hüthig, Heidelberg, [1]2001.

[7] Joachim von zur Gathen and Jürgen Gerhard, *Modern Computer Algebra*, Cambridge University Press, Cambridge, [1]1999.

[8] Jean-Charles Faugére, "A new efficient algorithm for computing Gröbner bases (F_4)," *J. Pure Appl. Algebra*, vol. 139, pp. 61–88, 1999.

[9] Jean-Charles Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)," Mora, Teo (ed.), ISSAC 2002. Proceedings of the 2002 international symposium on symbolic and algebraic computation, Lille, France, July 07–10, 2002. Pp. 75-83, ACM Press, New York, NY., 2002.

[10] Benedykt S. Rodanski and Marwan Hassoun, "Symbolic Analysis," in *The Circuits and Filters Handbook*, Wai-Kai Chen, Ed., chapter 24, pp. 753–779. CRC Press, Boca Raton, Florida, 2nd edition, 2002.

[11] Louis Weinberg, *Network Analysis and Synthesis*, McGraw-Hill, New York, 1962.

[12] Frank Garvan, *The Maple Book*, Chapman & Hall/ CRC, Boca Raton, FL, 2001.

[13] Thomas A. Feo and J. Scott Provan, "Delta-Wye Transformations and the Efficient Reduction of Two-Terminal Planar Graphs," *Operations Research*, vol. 41, pp. 572–582, 1993.

[14] Sven Domann, "Implementierung des Feo-Provan-Algorithmus zur Stern-Dreieck-Reduktion planarer Graphen," Studienarbeit, Technische Universität Braunschweig, 2003.

NOTE ADDED TO THE ELECTRONIC VERSION

In this electronic document keywords, ACM and MSC classifications have been added. (March 14th (Pi Day), 2009)

Eberhard H.-A. Gerbracht received a Dipl.-Math. degree in mathematics, a Dipl.-Inform. degree in computer science, and a Ph.D. (Dr. rer.nat.) degree in mathematics from the Technical University Braunschweig, Germany, in 1990, 1993, and 1998, respectively.

From 1992 to 1997, he worked as Research Fellow and Teaching Assistant at the Institute for Geometry at the TU Braunschweig. From 1997 to 2003 he was an Assistant Professor in the Department of Electrical Engineering and Information Technology at the TU Braunschweig. After a two-year stint as a mathematics and computer science teacher at a grammar school in Braunschweig and a vocational school in Gifhorn, Germany, he is currently working as tutor, and independent researcher in various areas of mathematics. His research interests include combinatorial algebra, C*-algebras, the history of mathematics in the 19th and early 20th century and applications of computer algebra and dynamical geometry to graph theory, calculus, and electrical engineering.

Dr. Gerbracht is a member of the German Mathematical Society (DMV), the Society of Computer Science Teachers in Lower Saxony within the Gesellschaft für Informatik (GI-NILL), and founding member of the society "Web Portal: History in Braunschweig - www.gibs.info".